Image Preservation Through PDF/A

Frank L. Walker and George R. Thoma
Lister Hill National Center for Biomedical Communications
National Library of Medicine
Bethesda, Maryland

Abstract

The preservation of image collections is a goal of many libraries and organizations. Among the considerations for long-term preservation is the choice of file format. An emerging file format for preservation is PDF/A, which is a proposed international standard for long-term electronic document preservation. Since this file format can be used for image preservation, there are a number of considerations to be followed for converting image files to PDF/A, and also for ultimately migrating from PDF/A to successor file formats. As a result of research and development in electronic document delivery and file conversion, an Internet-based file migration system called MyMorph has been created at the National Library of Medicine. MyMorph was developed as a file migration service that allows the bulk conversion of electronic documents to PDF in a manner that minimizes certain aspects of the migration cost. The user interface permits file migration to proceed in batch mode, requiring minimal user interaction regardless of the number of files converted. After more than two years of use, the system is being modified to facilitate the creation of PDF/A files from source files consisting of scanned images. This paper details the design decisions for extending MyMorph to create image-based PDF/A files.

Introduction

The preservation of image collections into the distant future depends on a number of factors, including the cost of collection acquisition, storage, file replication and file migration. Among the considerations for preservation is the choice of the file format, one that is likely to be maintained and supported in the future. An emerging file format for preservation is PDF/A, a proposed international standard for long-term electronic document preservation.[1] PDF/A defines a subset of PDF version 1.4 features that would be mandatory, allowed or prohibited.[2] Mandatory features include an XML-based metadata object that provides important metadata for the document, such as author, subject, creator, creation date and modification date. If a document is modified for preservation purposes in the future, the metadata object would also be modified to record the history of how and why the file was changed. While many features of PDF version 1.4 are allowed in PDF/A, others are restricted or prohibited. Also, PDF/A prohibits the inclusion of movies, sounds and JavaScript (executable code).

Two levels of conformance are possible for PDF/A: Level A and Level B. A file that meets Level A conformance is one that includes the full set of PDF/A features, including the document's logical structure and content text stream in natural reading order. These two features facilitate navigation of the document, and the possible migration of the PDF/A file to a future file format. The logical structure specifies the document's structure, which may include chapters, sections, footnotes, headers, figures and tables. This makes it possible for reader software to provide a user interface for easy navigation through the document. By maintaining the content text stream in natural reading order it is possible to present the text to physically impaired users. It also allows conversion of the content in its proper order for migration. Level A-conformant files also include a Unicode character map, which allows the retrieval of semantic properties of each character referenced in the file.

A PDF/A file meets Level B conformance if it satisfies all requirements except perhaps the logical structure, content text stream in logical reading order, and Unicode character map. A goal of PDF/A is to define an electronic document that retains its appearance over time. While an electronic document stored in PDF/A format may never change, the software tools for rendering it evolve over time. It is expected that a PDF/A file meeting Level A conformance will allow future software tools to render the document in its original appearance. Level A conformance also promises to allow migration of the file to future formats with little or no change in appearance. A file that meets Level B conformance should also retain its visual appearance as rendering tools change. However, the lack of logical structure, natural reading order of the text stream and Unicode character map may, in some cases, make it more difficult to migrate to future file formats.

Organizations that want to create image archives may choose PDF/A as the preservation file format. A primary reason for choosing PDF/A as the archival format is the widespread availability of the Acrobat reader on many computing platforms. The Acrobat reader provides a user

interface that makes it easy to view monochrome, grayscale and color images on a computer screen, and to print them. Images stored in many non-PDF formats, such as TIFF, can be converted to PDF/A without loss of information, and made viewable through the Acrobat reader. Since this reader is far more ubiquitous than viewers for most other image formats, PDF/A becomes a practical choice for image file format. This paper describes a prototype file migration system called MyMorph, developed at the National Library of Medicine, currently for creating PDF files via the Internet. It also describes the design considerations for modifying MyMorph to enable the creation of image-only PDF/A files.

Considerations for PDF/A Image Archives

There are a number of considerations to be taken into account when creating an image-only PDF/A archive. To migrate an existing image archive to PDF/A, or to create a PDF/A archive from newly scanned images, the first decision to be made is whether the PDF/A file should meet Level A or Level B conformance, denoted PDF/A-1A and PDF/A-1B respectively. It is probably not practical to achieve Level A conformance, since capture devices such as digital cameras or scanners that create raster images do not determine the logical structure, information flow, or Unicode character map. Adding these items manually would be costly. Level A conformance is primarily suitable for files containing text, rather than images. Files to be converted to PDF/A-1A would generally originate from word processors or other text-handling programs. This leaves PDF/A-1B the most practical alternative for a PDF/A image archive.

The next question is whether PDF/A-1B is satisfactory for maintaining an image archive for the very long term. Suppose an organization creates an image-only PDF/A archive today. Will it be usable decades from now? If so, what would be necessary for a future generation to use it? First, we must assume that during this time an archivist would replicate the archive on new media simply because physical media decay with time. In order to use the archive, it would be necessary to have two things. One is the specification describing how the archive was created. The second would be the specification describing how the images would be used, whether for migration to a new file format, or for display on a specific device. A future programmer would have to understand both the PDF/A specifications, and the specifications for the new format.

The specifications for PDF/A list a number of references that contain information necessary for understanding and using PDF/A files. The list of references is quite long, but some of the references are needed only for using PDF/A-1A files, not those that meet level B conformance. Table 1 is a typical list of documents needed by a future generation for migrating or using PDF/A-1B files that contain only images.

1. PDF Reference version 1.4, Adobe
2. Errata for above PDF Reference
3. ISO specification for PDF/A
4. Date and Time Formats, W3C
5. ICC.1:1998-09 File Format for Color Profiles
6. ICC.1A:1999-04 Addendum 2 to above
7. ISO/IEC 646, Information technology – ISO 7-bit coded character set for information exchange
8. Extensible Markup Language (XML) 1.0, 3rd Edition, W3C
9. RDF/XML Syntax Specification, W3C
10. XMP Specification, Adobe
11. Specifications for the internal storage format for images stored in the file.

Table 1. References for using PDF/A-1B Image Files

The task for migrating the archive at a distant point in the future could be formidable, and it could take a future archivist/programmer several years to create software for using the archive, if it does not exist. However, if programmers of the future are at least as capable of current programmers, then the task seems manageable.

MyMorph

One of the expectations for PDF/A is that software tools for creating, reading, rendering and verifying PDF/A files will become available once PDF/A becomes an ISO standard. One tool called MyMorph, developed as a file migration service at the Lister Hill National Center for Biomedical Communications, an R&D division of the National Library of Medicine (NLM), is currently used for creating PDF files via the Internet.[3,4,5] MyMorph is a technique for bulk file migration that is intended to minimize certain aspects of the migration cost. It is a Web-based approach that allows the conversion of a potentially large collection of electronic documents to PDF. It consists of client and server software that employ Simple Object Access Protocol (SOAP), a technology utilizing extensible markup language (XML) sent over the Internet via the Hypertext Transfer Protocol (HTTP). The MyMorph service relies on client software running on the user's computer to send files via SOAP to a system at NLM called DocMorph, which can convert more than fifty different file formats to PDF, and return the results to the user. After two years of beta testing, version 1.0 of the MyMorph client software was released in July 2004. MyMorph currently has more than 4,000 registered users, who have converted thousands of files to PDF. The beta test confirmed that users found the software easy to learn, and the conversion to be fast. The bulk of the user population consists of small libraries that use MyMorph as part of their document delivery operation, and they tend to convert up to ten files at a time, but there are also a few users who have done large-scale migration of document collections to PDF.

The software is designed to minimize the cost of migration for users in two ways. First, since MyMorph is freely available, anybody with a Windows-based computer and access to the Internet can use it. Second, the user interface permits file migration to proceed in batch mode with minimal user interaction regardless of whether a single file or hundreds at a time are converted. MyMorph is available through the DocMorph Web site at http://docmorph.nlm.nih.gov/docmorph.

Figure 1 shows the architecture of the MyMorph/DocMorph system. There are six processors comprising the DocMorph server, with one processor designated as the Permission computer, and the five others as Worker computers. Users can access the system via Web browsers or MyMorph clients running on their computers. The browser or MyMorph client is granted permission to use the system by the Permission computer, which then routes the browser or MyMorph client to a Worker computer where the files are converted. DocMorph provides five conversion functions to Web browser users, including creating TIFF files, splitting multi-page TIFF files, creating PDF files, extracting text from files, or "reading aloud" the file using speech synthesis. MyMorph users are limited to the creation of PDF files. The advantage of using MyMorph over a browser for creating PDF files is that it requires very little user interface time, and this time is independent of the number of files to be converted.

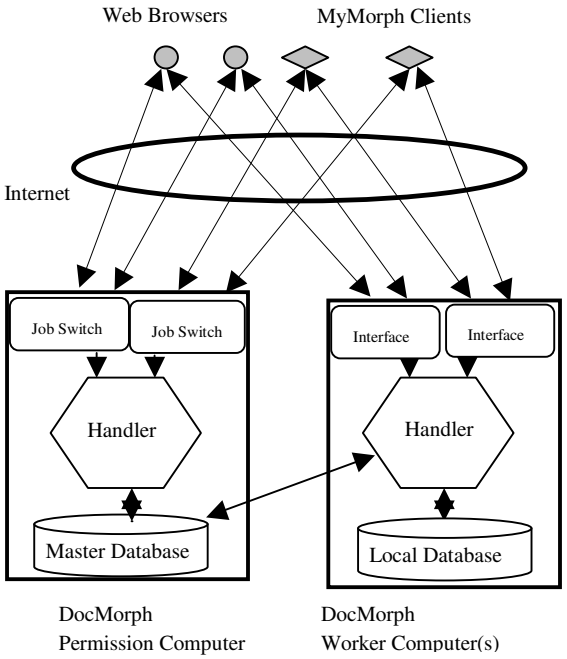

Figure 1. DocMorph/MyMorph System Architecture

Since MyMorph is designed for converting potentially large collections of files to PDF, it is a candidate for creating archives of PDF/A files. After a thorough review of the proposed specifications for PDF/A, it was determined that MyMorph would be suitable for modification to create PDF/A-1B files from image files. While the system handles many types of files, including image files and word processing files, the immediate development will focus on image-only files. These file formats include BMP, DCM, DCX, DIB, FPX, GIF, ICO, JBIG, JPEG, MTV, PDB, PDM, PCD, PCX, PCM, PIX, PNG, PNM, PPM, PSD, RAS, TIFF, XBM and Ariel system GEDI files.

The DocMorph software consists mainly of a combination of in-house designed C++ code, and two software packages, ImageMagick and Ghostscript, freely available on the Internet for a variety of computer platforms. ImageMagick is a collection of image processing utilities capable of handling over 90 different image file formats.[6] DocMorph uses ImageMagick's *Convert* utility to process most image file formats submitted to the system. As an example, it uses *Convert* to change a BMP format file to a color RGB format that is stored inside the PDF file.

Design Considerations for PDF/A

Four primary considerations were taken into account for redesigning the MyMorph/DocMorph system to create PDF/A-1B files from image files: image downsampling, lossy compression, color spaces and metadata. Because downsampling reduces the image resolution, the PDF/A specification recommends that images not be downsampled during conversion to PDF/A. Choice of a suitable compression algorithm for storing the image favors selecting lossless algorithms over lossy ones. Colors in PDF/A are specified in a device-independent manner to increase the flexibility for conversion to future file formats. Finally, PDF/A allows a complete description and history of the file to be stored within its XML-based metadata object. These four considerations are discussed next.

Image downsampling is addressed in an appendix of the PDF/A specification. Although the specification does not prohibit image downsampling, it recommends that, as a best practice, images should not be downsampled when converted to PDF/A. The specification favors image quality over image size, and stresses the importance of preventing loss of information. Downsampling, the process of reducing the image size by eliminating data, is normally done to scale an image for display on a lower resolution device. If, for example, an image consists of bits 1, 2, 3, 4, 5, 6, and so on, then downsampling by a factor of 4 results in an image with bits 1, 5, 9 and so on. Other methods of downsampling may include linear interpolation and use of low pass filters with subsampling. While downsampling reduces the size of the image, it may reduce the image quality on high-resolution display devices. Even though current display hardware may not be capable of displaying the full resolution of an image, a future generation device may be capable of a more legible display. Since retaining image quality is of prime

importance in preservation, downsampling is not recommended. Preservationists still have to deal with image size and the capability of storage media for archiving an image collection, but trends in recent years tend to predict that storage density will continue to increase for some time to come. The DocMorph conversion software was redesigned to ensure that it did not downsample images during conversion to PDF/A.

In addition to discouraging downsampling, the PDF/A specification also recommends against using lossy compression when converting images to PDF/A. Compression can be lossless, as in Group 4 compression for black and white images, or it can be lossy, as in JPEG. The PDF version 1.4 specification allows images to be stored within the PDF file using several image compression methods: JPEG compression for color and grayscale images; run-length compression, ITU-T Group 3, ITU-T Group 4 or JBIG2 compression for monochrome images; and Lempel-Ziv-Welch (LZW) and Flate compression for color, grayscale or black and white images. The PDF/A specification excludes LZW compression, since up until recently its use had legal ramifications. JPEG compression always results in loss of image quality, although the quality can be determined at the time the image is stored in JPEG format. Images in JPEG format tend to lose sharpness at edge boundaries (for example, a row of black pixels adjacent to a row of white pixels). Sharp edges tend to be blurred unless the image quality is selected to be high, i.e., compression ratio is low, which offsets the advantage of compression. To prevent loss of information during the conversion process to PDF/A, the specification's "best practices" section would rule out JPEG compression. This leaves run-length compression, Group 3, Group4, JBIG2 and Flate as the choices for image compression, since they are all lossless. For monochrome images DocMorph uses Group 4 compression. Group 4 uses a two-dimensional compression algorithm that results in better compression than either run-length compression or Group 3 compression. Another possibility for compression of monochrome images is JBIG2, which promises to yield three to five times better compression than Group 4. Since the ImageMagick software does not handle JBIG2, this technique is not currently included in DocMorph.

The only viable method for compressing color images in PDF/A is Flate. Newer versions of PDF allow lossless JPEG2000 compression, but since this recent standard is not available in version 1.4, it is also not permitted in PDF/A. Flate, also called ZIP, is a compression method that works well on images with large areas of single colors or repeating patterns. Flate uses a combination of the LZ77 algorithm (a non-proprietary precursor to LZW) and Huffman coding.[7,8] For the PDF/A development, DocMorph was modified to use Flate compression to compress grayscale and color images using the freely available ZLIB library of software compression routines.[9]

A third consideration for modifying DocMorph to create PDF/A files is the use of color spaces. A color space, which can be device-dependent or device-independent, is the range of colors in which color images can be represented. A device-dependent color space is a way of describing the colors in an image that allows it to be displayed on a specific device, whether a hardcopy printer or softcopy monitor. While PDF version 1.4 supports three types of device-dependent color spaces (*DeviceGray, DeviceRGB and DeviceCMYK*), four types of device-independent color spaces (*CalGray, CalRGB, Lab and ICCBased*), and four special color spaces (*Index, Pattern, Separation and DeviceN*), PDF/A restricts the color space to be one of the four device-independent color spaces. Using a device-independent color space allows more flexibility in converting the PDF/A file to future file formats, and for display on a variety of devices. If an *ICCBased* color space is used, then PDF/A requires the color space to be embedded in the PDF/A file. The International Color Consortium, or ICC, is an industry consortium of about 70 member organizations whose goal is to promote use of a vendor-neutral cross-platform color management system. It has developed a number of ICC Profiles that contain transforms to and from device color spaces to the *Profile Connection Space*, which is device-independent. Because of the general-purpose nature of DocMorph and MyMorph, the user is not required to specify how the colors in the submitted file are rendered, i.e., whether the colors are device-dependent (such as RGB or CMYK), or device-independent. For initial development, it is assumed that color images have been rendered using sRGB, which is the default color space for multimedia applications and images posted on Web sites, and was developed through work at Microsoft and Hewlett-Packard.[10] Images rendered using sRGB can be displayed on most uncalibrated monitors fairly accurately. It may be incorrect to assume that images submitted to the system are rendered as sRGB. Because this may cause some colors to be displayed inaccurately, our design will allow the user to choose the color space prior to submitting the files for conversion.

The chief modification to DocMorph to accommodate the sRGB color space consists of using an OutputIntents dictionary located in the Catalog object that has GTS_PDFA1 as the value of its S key, and the sRGB profile stream as the value of the DestOutputProfile key. Figure 2 is a typical example of the OutputIntents array and dictionary.

```
28 0 obj      % ICC Profile Stream
<<
/N 3
/Length 3144
>>
stream
00 00 0C 48 4C …
endstream
endobj

/OutputIntents
[
<<
/Type /OutputIntent
/S /GTS_PDFA1
/DestOutputProfile 28 0 obj
/OutputConditionIdentifier (sRGB)
/RegistryName (http://www.color.org)
>>
]
```

Figure 2. Example OutputIntents Array and Dictionary

The final consideration for modifying DocMorph to handle PDF/A files is metadata. The requirement for PDF/A files to contain a metadata object is critical for preservation, since the metadata object includes pertinent information about who created the file, when it was created, when it was modified, and the reasons for modification. Thus the metadata object contains the complete history of the file. Prior to its current redesign, DocMorph created PDF files containing a Document Information Dictionary object, consisting of the name of the creator and date of creation. The PDF/A specification requires synchronization between the metadata object and Document Information Dictionary, so that the information in the latter must also exist in the metadata object. DocMorph was modified to satisfy this requirement, so that when it creates the PDF/A file the entries in the Document Information Dictionary and the metadata object are consistent. To ensure that it creates a valid metadata object, DocMorph software uses the XMP Toolkit from Adobe Systems, Inc. This toolkit provides source code for implementing the Extensible Metadata Platform, a specification from Adobe that provides an XML-based framework for working effectively with metadata.[11] DocMorph was modified to create a minimal metadata object, containing the *CreatorTool*, *Producer*, *CreateDate*, *ModifyDate*, *DocumentID*, *part* and *conformance* fields.

In addition to the four considerations for DocMorph modification, several minor modifications were necessary, one of which was to include the ID parameter in the file trailer dictionary. The ID parameter uniquely identifies the file, and consists of an array of two uniquely created strings that are initially identical. While the first string is fixed, the second string changes whenever the file is modified. The new value of the second string is based on the file's contents at the time the file is modified. This is what a typical ID looks like when first created:

[<83f48392eefa8bc9028793748495833a><83f48392eefa8bc9028793748495833a>]

The Next Steps

It is expected that tools for creating and validating PDF/A files will become available after PDF/A becomes an accepted standard. We are waiting for PDF/A validation tools to appear before finalizing our modifications to MyMorph/DocMorph. This will allow us to verify that the files created by the system are valid PDF/A files, and conform to PDF/A-1B. We are also considering giving the user options for selecting the color space to be used in the file. Finally, we plan to explore modifying the system to store black and white images in JBIG2 format within PDF/A files, which should result in three to five times better compression than Group 4 compression. This would be of interest to image preservationists concerned with collection size.

Summary

PDF/A is an emerging file format specification for preserving electronic documents. It permits two levels of conformance, each of which promises to provide long-term preservation. An analysis of the specification reveals that image files preserved through PDF/A are best suited for level B conformance. An existing system for migrating files to PDF via the Internet called MyMorph is being modified to permit migration of image files to PDF/A files with level B conformance. The design factors considered for system modification include image downsampling, lossy compression, color spaces and metadata.

References

1. ISO/DIS 19005-1, Document management – Electronic document file format for long-term preservation – Part 1: Use of PDF 1.4 (PDF/A-1).
2. Adobe Systems Incorporated, PDF Reference: Adobe Portable Document Format, Version 1.4, Addison-Wesley, Boston, 3rd edition (2001).
3. Frank L. Walker and George R. Thoma, Web-based Document Image Processing. Proceedings of IS&T/SPIE Conference on Internet Imaging, San Jose, California, 268-277 (January 2000).
4. Frank L. Walker and George R. Thoma, A SOAP-Enabled System for an Online Library Service. Proceedings of InfoToday 2002. Medford N.J.: Information Today, 320-329 (2002).
5. Frank L. Walker and George R. Thoma, A Web-Based Paradigm for File Migration. Proceedings of IS&T's 2004 Archiving Conference. SpringField, VA: IS&T: The Society for Imaging Science and Technology, 93-97 (April 2004).

6. ImageMagick is available at http://www.imagemagick.org.
7. RFC 1950, ZLIB Compressed Data Format Specification Version 3.3, http://www.faqs.org/rfcs/rfc1950.html.
8. RFC 1951, DEFLATE Compressed Data Format Specification, Version 1.3, available at http://www.faqs.org/rfcs/rfc1951.html.
9. ZLIB is available at http://www.gzip.org/zlib.
10. Mary Nielsen and Michael Stokes, The Creation of the sRGB ICC Profile. IS&T/SID Sixth Color Imaging Conference: Color Science, Systems and Applications. Scottsdale, Arizona, 253-257 (November 1998).
11. Adobe Systems Incorporated, XMP Specification (2004).

Biography

Frank L. Walker received his B.S. and M.S. degrees in electrical engineering from the University of Maryland. Since he joined the National Library of Medicine in 1979, he has designed, developed, performed research and published a number of papers on computer systems utilizing electronic imaging, primarily for the purpose of electronic document storage, retrieval, transmission and use. His current interest is in developing software and systems for improving the delivery and use of biomedical library information.

George R. Thoma is a Branch Chief at an R&D division of the U.S. National Library of Medicine. He directs R&D programs in document image analysis, biomedical image processing, animated virtual books, and related areas. He earned a B.S. from Swarthmore College, and the M.S. and Ph.D. from the University of Pennsylvania, all in electrical engineering. Dr. Thoma is a Fellow of the SPIE, the International Society for Optical Engineering.